FIG. 1

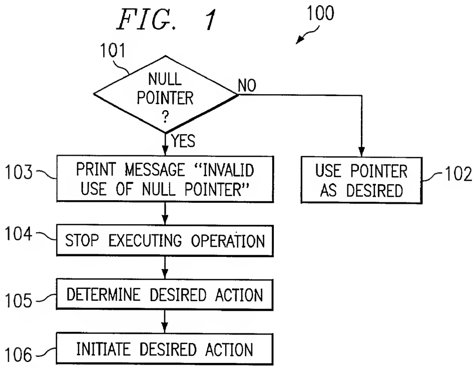

FIG. 2

```
201   #include <iostream.h>

203   template <class C>
204   class SafePtr {        207        208
205     public:                                   210
206       SafePtr (C *p) : ptr(p) { }
209       operator C *() const { return ptr; }
211       C *operator-> () const {
212           if (!ptr) {
202               cerr << "\nAttempted indirection of "
                       << "NULL pointer.\n" ;
                  exit(1) ;          // or throw an exception
              }
213           return ptr;
          }
214       C operator* () const { return *operator->() ; }
215     private:
          C *const ptr;
      } ;
```

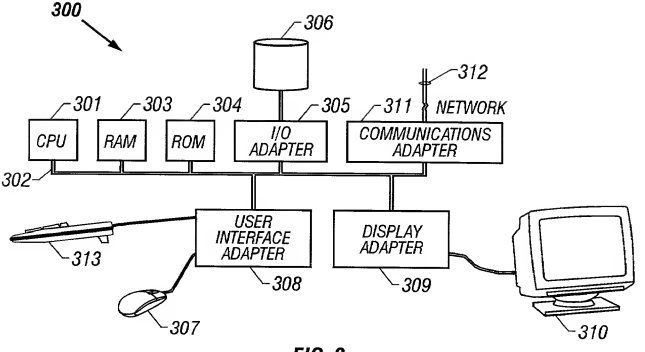
300
306
312
301
303
304
305
311
NETWORK
CPU
RAM
ROM
I/O ADAPTER
COMMUNICATIONS ADAPTER
302
313
USER INTERFACE ADAPTER
DISPLAY ADAPTER
308
309
307
310

FIG. 3